

EMBEDDED CO-PROCESSOR ARCHITECTURE FOR CMOS BASED IMAGE ACQUISITION

J. Dubois, M. Mattavelli
Swiss federal institute of Technology
Signal Processing Laboratory 3 (LTS-3)
CH-1015 Lausanne, Switzerland
Email: julien.dubois@epfl.ch

ABSTRACT

This paper describes a new co-processor architecture designed for CMOS sensor imaging. The co-processor unit is integrated into the image acquisition loop so as to exploit the full potential of CMOS selective access imaging technology. The processing features of the co-processor are functional to the specific acquisition process of CMOS sensors (random region acquisition, variable image size, variable acquisition modes line/region based, multi-exposition images). Moreover, although built with pipelined or parallel HW processing modules, the co-processor architecture has been designed so as to obtain a unit that can be configured on the fly, in terms of type and number of chained processing, during the image acquisition process that is defined by the application. Simulated performances based on a FPGA implementation, are reported and compared to classical image acquisition systems based on PC platforms.

1. INTRODUCTION

For very high-speed image-processing applications a fast and adaptive image acquisition stage is very often a key feature to achieve real-time performance and thus satisfy the application requirements. The large volume of data to be transferred to the central processing unit from the image sensor is often the system bottleneck in terms of performance, the response time of the system is too slow because the transfer time is too large, or in terms of costs, the (large) bandwidth required is too costly in terms of equipment and interfaces. Although we assist to a continuously increasing of processing performances and at the appearance of fast bus interfaces, the availability of high-speed high-resolution sensors pushes the performance to higher limits so as to cover new demanding applications.

So as to reach real-time, the co-processing approach has been often used in the last years. Some approaches presented in literature are based on hardware co-processing designs specifically dedicated to single

applications [1][2][3]. The reported performance improvements are relevant and range up to a factor several hundreds when compared to the base architecture without co-processor. Other authors have proposed generic systems characterized by the possibility to implement different algorithms on a co-processing based architecture [4]. The performance of such more flexible implementations in terms of speed-ups ranges up to several tens factors for some specific processing. In the class of “generic” co-processor units, only few authors have mentioned the tentative to control the image acquisition stage simultaneously with the processing stage. Gorgon has proposed a co-processor unit to control the acquisition stage of Charge Coupled Devices (CCD) sensor [5]. Jung et al. presented a pre-processing unit to control CMOS sensor [6], but the implemented functionality operates only on the specific image corrections used to compensate physical limitation of the CMOS sensor. Although CMOS sensor present very attractive features, no works presented in literature have shown that acquisition can be adapted to the processing providing a processing stage similar to the one we can find in “retina” sensor approaches [7].

This paper describes a co-processor unit design (COP) providing an interface for the full control of the CMOS sensor acquisition process driven from the main application CPU. The main processor and the co-processor are respectively in charge of the high-level tasks, the acquisition and processing decision imposed by the application, and the lower-level tasks, characterized by high level of processing regularity and parallelism. The co-processor implementation is based on a standard Field-Programmable Gate Array (FPGA) technology. The main results achieved in this work are twofold. The first interesting result is that relevant speed-up factors are obtainable for reconfigurable processing modules, thus providing enough flexibility in term of choice of processing and in terms of acquisition mode defined on the fly by the application itself (selection and preprocessing of any kind of area of interest). The second is that such on the fly adaptation of the acquisition mode yields a further bandwidth reduction for the transfer of the image data to the central CPU. This feature represents

for some application a further speed-up in the overall system performance in terms of reduction of processing or increase of the achievable acquisition/processing frame rate.

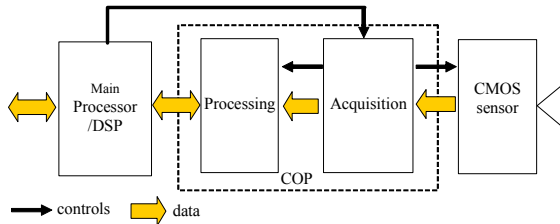


Fig. 1. Block diagram of the co-processor based architecture.

The co-processor commands and the data are transferred between the main CPU and the by a common bus. The command word bandwidth is negligible compared to the image data volume. The co-processor operations are determined by command received by the main processor together with the acquisition commands.

The paper is organized as follows: section 2 presents how the inclusion of processing into the acquisition loop enables to exploit the features and innovations of CMOS based imaging. In section 3 the co-processor architecture is presented and his features are discussed in detail. Finally, the performance of the co-processor architecture obtained by simulations, are reported in Section 4 and compared to a classical image acquisition and processing scheme.

2. CO-PROCESSOR INTO PROCESSING ALGORITHM/ACQUISITION LOOP

The integration of a co-processing element into the image acquisition loop of a CMOS sensor has very interesting features. Standard CCD based image systems are synchronous and require that the full image is downloaded before proceeding to a new acquisition. CMOS sensors are much more flexible because not only are intrinsically asynchronous, but are also capable of performing image acquisitions on limited section of the sensor up to the acquisition of single pixels. For several applications such flexibility can be successfully exploited so as to reduce the data transfer to the central CPU thus considerably reducing the necessary data bandwidth and as consequence the overall processing requirement of the application that has just to process a limited portion of the original image. The key to achieve such results is to be able to provide to the main application the necessary information to adapt the acquisition stage without the need to transfer the full image to the central CPU. In other words, CMOS imaging can achieve:

- a selective image acquisition stage depending on the image content itself and on the requirements of the application,
- a relevant reduction of the data volume to be transmitted to the central CPU once the selective acquisition stage has been activated.

The condition for which such features can be achieved is that a “co-processing” element is inserted in the image acquisition loop driven by the “high level” application. In such architecture the “co-processing” unit beside the control of the acquisition stage becomes naturally in charge of the standard low-level repetitive tasks such as filtering, de-noising, binarisation, etc. In fact the full control of the acquisition stage enables the right control of the pre-processing tasks usually performed at the level of the central CPU or high-level application. For instance, the “instructions” for a selective image acquisition stage, i.e. an acquisition stage for which only a (small) portion of the image that presents certain features need to be “acquired” and transmitted to the central CPU for further high level processing are handled by the “co-processor” accessing directly the CMOS sensor itself in an asynchronous manner. At this point also the processing associated to the specific feature “found” in the image can efficiently be implemented at the “co-processor” level. Then only the “selected” image portion already pre-processed and/or pre filtered is transferred to the central CPU unit. The co-processing task schedule can be selected on the fly depending on the acquisition commands and is adapted to the acquisition form that is region/pixel based. With this architectural approach, only the CMOS sensor is providing the input image, thus the overall system results very similar to a “retina” [7]. By this approach the necessary data bandwidth can be drastically reduced eliminating in most of the cases the major system limitation. An example of achievable performance for some classical pre-processing stage is provided in section 4. The main processor, freed from image acquisition and pre-processing tasks can then be used for further processing and/or high-level algorithms defined by the specific application.

The challenging aspects of the co-processor design are mainly related to the variable acquisition mode (i.e. input image format and layout) with the optimization of the associated access bandwidth with the CMOS sensor and with the degree of flexibility of the number order and nature of possible pre-processing stages that can be associated at each acquisition mode. In the examples of co-processing performance provided in this paper the acquisition command word set generated by the processor are constituted essentially by two parts: the processing order with the parameters and the acquisition part. Each acquisition field is coded on 16 bits. Many different acquisition modes are then available. In all modes, a

window can be selected in the full-range image, the size and the integration time are defined moreover a sub-sampling (on Y and X) can also be specified. In simple multi-exposition mode, the same window is acquired several times or periodically and the delay between two acquisitions can be defined. Moreover in the tracking multi-exposition mode, the window can also be translated. Such modes permit to create a “sub-image” image by row or column accumulation when the sensor is used as line sensor even with lines varying their position during the acquisition itself.

3. CO-PROCESSOR DESIGN

As mentioned in the previous section the essential problem of the co-processor design for CMOS image sensors is the trade-off between processing efficiency and flexibility required to exploit the CMOS potential features. The COP architecture is essentially constituted by the following functional blocks: a processor interface (bus interface), a bus bridge, a command controller, a processing controller, a processing structure and finally a CMOS sensor interface (Fig. 2).

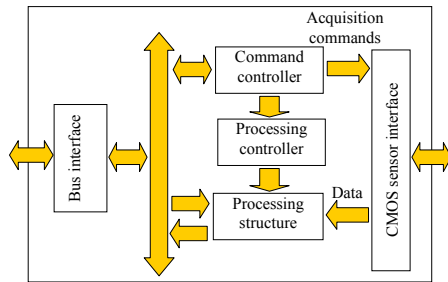


Fig. 2. Block diagram of the COP architecture

The command controller receives the acquisition commands, the processing commands from the main application, and then the information for the acquisition stage is transferred to the CMOS sensor interface and the processing command to the processing controller. The task scheduling is controlled by the processing controller and is executed by the processing structure unit configured according to the received commands. The data and image portions, provided by the main CPU and used by the co-processor for the actual processing tasks, are transferred to the processing structure via the bus bridge and via the processing controller. This feature enables to implement a true co-processing stage and not a simple pre-processing.

The possibility to adapt the number and nature of the processing and to operate on variable size/shape images is provided by the flexibility of the processing structure unit. In essence it is constituted by five different components (Fig. 3): CONTROL_MEM is in charge of the main memory, CONTROL_PRO is in charge of the

processing control, the processing modules, the system control and the FIFO is in charge of the temporary storage. Such architecture permits several possibilities for the data flow control (Fig. 3). The input data, provided by CMOS sensor and by the processor, are referred in Fig.3, respectively with the number 1 and 3. There is no FIFO in 1 since there is a memory in the CMOS interface. The broadcasting nets referred with 2, 4 and 5 permit to copy the data and transfer them on each output branches. The copy is specified for each net by the command word. The nets referred as 2 permits to transfer the input image without processing. The nets 4/5 permit to transfer the result image between two processing, simultaneously with the data loading/the result reading. The processing structure unit can be configured to adapt its processing in function of the acquisition mode and in function of the high level application via software.

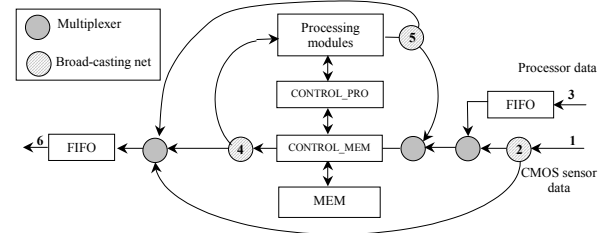


Fig.3. Processing structure of the COP architecture

The current acquisition data has to be stored into an internal memory to permit the pre-processing stage. Several types of pre-processing require a pixel neighbourhood for each pixel process. A common way to operate is to use video line to store few image rows. Unfortunately, such solution is not possible because the sub-image size is not fixed. In the architectural solution presented here, an internal cache memory is associated at each processing. Consequently, the processing flow might not be synchronised with the output data flow of the memory MEM. Such solution enables to decrease the number of accesses to MEM. The size and features of the cache are defined to match the selected processing.

The processing modules are sharing the same input and output busses that are connected to the bi-directional main memory bus. So as to store the results in the same memory, the input data enables to cascade the processing or to apply the same processing several times.

4. EXAMPLE OF ACHIEVEABLE PERFORMANCE

The co-processor has been simulated and implemented on FPGA Virtex-II XC2V1500. The implementation required 1600 logical elements (slices) without the CMOS sensor interface and the processing. 2 MBytes of

main memory are added as external memory. Three different processing types have been implemented:

- a median filter on different basic kernels (1x3, 1x5, 3x3),
- a local adaptive binarisation (Niblack algorithm) with a neighbourhood of 8x8 or 16x16 pixels [8]
- a binary pattern recognition based on block matching with 32x32 and 64x64 pattern size.

The performances obtained by the co-processor architecture are reported Table 1. The required hardware-resources are reported in Table 2.

<i>Median filter</i>	<i>1x3/1x5</i>	<i>Local adaptive binarisation</i>	<i>M8x8</i>	<i>Binary pattern matching</i>	<i>Pattern size 32x32</i>
512x512	2.61	512x512	10.24	512x512	18.40
256x256	0.65	256x256	2.50	256x256	4.03
128x128	0.16	128x128	0.60	128x128	0.75
<i>Median filter</i>	<i>3x3</i>	<i>Local adaptive binarisation</i>	<i>N16x16</i>	<i>Binary pattern matching</i>	<i>Pattern size 64x64</i>
512x512	5.22	512x512	19.84	512x512	32.1
256x256	1.30	256x256	4.69	256x256	5.92
128x128	0.32	128x128	1.04	128x128	0.67

Table 1. Processing performances (ms) for different image sizes and neighbourhood/pattern sizes.

<i>Median filter</i>	<i>1x3/1x5 3x3</i>	<i>Local adaptive binarisation</i>	<i>N8x8, N16x16</i>	<i>Binary pattern matching</i>	<i>32x32 64x64</i>
Number of slices	265 313	Number of slices	1286	Number of slices	2407 3021

Table 2. Hardware processing resources.

Used in the pre-processing stage, the local adaptive binarisation permits to reduce the bandwidth to the central CPU like a retina sensor can perform. For example, a 1024x1024 full-range image requires 1 Mbytes to be stored but the binarised image only 1 Mbits. If an area can be selected in the full-range image, for example a 256x256, the result image size would reduce at 64 Kbits. This process permits to gain a factor 64 on the bandwidth.

A comparison has been done between the performance obtained by the co-processor architecture (COP) and a PC, Bi-Xeon 1.7 GHz, 256 Mo Ram, Rambus 800 MHz (2x400MHz). The performance results reported in Table 3 do not consider camera frame-grabber transfer time. The comparison permits to show that, besides the achieved speed-up factor up to a factor of 5 that would certainly result higher considering the frame-grabber transfer time, the central CPU in the co-processor approach is fully available for further processing. Moreover, when a bandwidth reduction is possible by

means of adaptive acquisition the co-processor approach provides much higher speed-up gains.

<i>Processing</i>	<i>PC (Mpixel/s)</i>	<i>COP (Mpixel/s)</i>
Median 1*3	41	100
Median 1*5	28	100
Median 3*3	27	50
Niblack 8*8	5	25
Niblack 16*16	4	14

Table 3. Processing comparisons.

5. CONCLUSION

Despite the availability of higher and higher speed PC processors, the implementation of relatively simple co-processor systems expressly conceived for CMOS image sensors and inserted in the acquisition loop has shown several advantages. Very high processing speed and reduced image data bandwidth are achievable maintaining at the same time a high degree of flexibility in the pre-processing stage for the different acquisition modes specific of CMOS imaging.

[1] B. Bosi, G. Bois, Y. Savaria, "Reconfigurable pipelined 2-D convolvers for fast digital signal processing," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, Volume 7, Issue 3, pp. 299–308, Sep 1999

[2] C.W. Murphy, D.M. Harvey, "Reconfigurable hardware implementation of BinDCT," *Electronics Letters*, Volume: 38 Issue 18, pp. 1012–1013, Aug 2002

[3] N.W. Bergmann, Yuk Ying Chung, "Video compression with custom computers," *IEEE Transactions on Consumer Electronics*, Volume 43, Issue 3, pp. 925–933, Aug 1997

[4] C. Hinkelbein, et Al., "Pattern recognition algorithms on FPGAs and CPUs for the ATLAS LVL2 trigger," *IEEE Transactions on Nuclear Science*, Volume 47, Issue 2, pp 362–366, Apr 2000

[5] M. Gorgon, J. Pryzybylo, "FPGA based controller for heterogenous image processing system," *Proceedings Euromicro Symposium on Digital Systems Design 2001*, pp. 453–457, 2001

[6] Yun Ho Jung, Jae Seok Kim, Bong Soo Hur, Moon Gi Kang, "Design of real-time image enhancement preprocessor for CMOS image sensor," *IEEE Transactions on Consumer Electronics*, Volume 46 Issue 1, pp. 68–75, Feb 2000

[7] F. Paillet, D. Mercier, T.M. Bernard, "Second generation programmable artificial retina" *Proceedings Twelfth Annual IEEE International ASIC/SOC Conference*, pp. 304–309, 1999

[8] O.D. Trier, A.K. Jain, "Goal-directed evaluation of binarization methods," *IEEE Transactions on PAMI*, Volume 17, Issue 12, pp. 1191–1201, Dec 1995